import java.util.ArrayList;

import java.util.HashMap;

import java.util.List;

import java.util.Map;

import java.util.Scanner;

class User {

private String username;

private String password;

public User(String username, String password) {

this.username = username;

this.password = password;

}

public String getUsername() {

return username;

}

public String getPassword() {

return password;

}

}

class Reservation {

private String pnr;

private String trainNumber;

private String classType;

private String dateOfJourney;

private String from;

private String to;

public String getPnr() {

return pnr;

}

public String getTrainNumber() {

return trainNumber;

}

public String getClassType() {

return classType;

}

public String getDateOfJourney() {

return dateOfJourney;

}

public String getFrom() {

return from;

}

public String getTo() {

return to;

}

public Reservation(String pnr, String trainNumber, String classType, String dateOfJourney, String from, String to) {

this.pnr = pnr;

this.trainNumber = trainNumber;

this.classType = classType;

this.dateOfJourney = dateOfJourney;

this.from = from;

this.to = to;

}

// Getters for reservation details

}

public class OnlineReservationSystem {

private static Map<String, User> users = new HashMap<>();

private static List<Reservation> reservations = new ArrayList<>();

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.println("Welcome to the Online Reservation System");

// Create a sample user for demonstration

users.put("user123", new User("user123", "password123"));

while (true) {

System.out.println("1. Login");

System.out.println("2. Reservation System");

System.out.println("3. Cancellation Form");

System.out.println("4. Exit");

System.out.print("Select an option: ");

int option = scanner.nextInt();

switch (option) {

case 1:

loginUser(scanner);

break;

case 2:

if (currentUser == null) {

System.out.println("Please login first.");

} else {

// Implement the reservation form

// Capture user details and create a reservation

createReservation(scanner);

}

break;

case 3:

if (currentUser == null) {

System.out.println("Please login first.");

} else {

// Implement the cancellation form

// Capture PNR number and display reservation details

cancelReservation(scanner);

}

break;

case 4:

System.out.println("Exiting the Online Reservation System.");

System.exit(0);

default:

System.out.println("Invalid option.");

}

}

}

private static User currentUser;

private static void loginUser(Scanner scanner) {

System.out.print("Enter your username: ");

String username = scanner.next();

System.out.print("Enter your password: ");

String password = scanner.next();

User user = users.get(username);

if (user != null && user.getPassword().equals(password)) {

currentUser = user;

System.out.println("Login successful. Welcome, " + currentUser.getUsername() + "!");

} else {

System.out.println("Login failed. Please try again.");

}

}

private static void createReservation(Scanner scanner) {

System.out.println("Reservation Form");

System.out.print("Enter your PNR number: ");

String pnr = scanner.next();

// You would typically retrieve the train details from a database

// For this example, we'll ask for basic details

System.out.print("Enter train number: ");

String trainNumber = scanner.next();

System.out.print("Enter class type: ");

String classType = scanner.next();

System.out.print("Enter date of journey: ");

String dateOfJourney = scanner.next();

System.out.print("Enter from (place): ");

String from = scanner.next();

System.out.print("Enter to destination: ");

String to = scanner.next();

Reservation reservation = new Reservation(pnr, trainNumber, classType, dateOfJourney, from, to);

reservations.add(reservation);

System.out.println("Reservation created successfully!");

}

private static void cancelReservation(Scanner scanner) {

System.out.print("Enter the PNR number for cancellation: ");

String pnrToCancel = scanner.next();

boolean foundReservation = false;

int indexToRemove = -1;

// Search for the reservation with the provided PNR number

for (int i = 0; i < reservations.size(); i++) {

Reservation reservation = reservations.get(i);

if (reservation.getPnr().equals(pnrToCancel)) {

foundReservation = true;

indexToRemove = i;

System.out.println("Reservation Details:");

System.out.println("PNR Number: " + reservation.getPnr());

System.out.println("Train Number: " + reservation.getTrainNumber());

System.out.println("Class Type: " + reservation.getClassType());

System.out.println("Date of Journey: " + reservation.getDateOfJourney());

System.out.println("From: " + reservation.getFrom());

System.out.println("To: " + reservation.getTo());

break;

}

}

if (foundReservation) {

System.out.print("Confirm cancellation (yes/no): ");

String confirmation = scanner.next();

if (confirmation.equalsIgnoreCase("yes")) {

reservations.remove(indexToRemove);

System.out.println("Reservation with PNR " + pnrToCancel + " has been canceled.");

} else {

System.out.println("Cancellation not confirmed. Reservation remains active.");

}

} else {

System.out.println("Reservation with PNR " + pnrToCancel + " not found.");

}

}

}